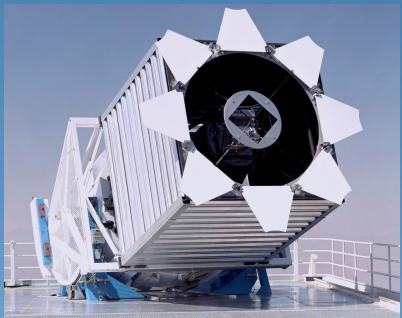


Week 1: Introduction and warm-up with astroML

LSST



SDSS



Astroinformatics, Fall 2018
Astronomical Observatory Belgrade

Outline

- Syllabus, etc
- Motivation for astrostatistics and this class
 - ever increasing data volume and complexity
 - sophisticated analysis, need for reproducibility
 - open-source approach
 - generally useful tools
- astroML
 - what is it?
 - what can you do with it?

• Syllabus and all that Everything is on GitHub:

The screenshot shows a Mozilla Firefox browser window with multiple tabs open. The active tab is 'astromundus/astroinfo2018: for students Fall 2018 - Mozilla Firefox'. The page displays the GitHub repository for 'astroinfo2018'. The repository summary shows 8 commits, 1 branch, 0 releases, and 1 contributor. The 'Code' tab is selected. The repository contains files like 'syllabus', 'lectures-notebooks', and 'README.md'. The 'README.md' file content is displayed as 'astroinformatics'.

astromundus/astroinfo2018: for students Fall 2018 - Mozilla Firefox

File Edit View Bookmarks Tools Help

AOB Webmail:: Inbox X Bridž Savez Srbije X B92 - Internet, Radio i TV X astromundus/astroinfo2018 X Week-1-1 X Saving changes | Atlassian X +

GitHub, Inc. (US) https://github.com/astromundus/astroinfo2018 133% ... Search

Search or jump to... Pull requests Issues Marketplace Explore

astromundus / **astroinfo2018** Watch 0 Star 0 Fork 0

Issues 0 Pull requests 0 Projects 0 Wiki Insights Settings

for students Fall 2018 Edit

Manage topics

8 commits 1 branch 0 releases 1 contributor

Branch: master New pull request Create new file Upload files Find file Clone or download

darkojev syllabus Latest commit 71ba4d0 3 minutes ago

lectures-notebooks lectures 23 minutes ago

syllabus syllabus 3 minutes ago

README.md first cmt 19 hours ago

README.md

astroinformatics

ASTROINFORMATICS

Introduction to AstroStatistics and Big Data in Astronomy

Darko Jevremović

Astronomical Observatory Belgrade, Fall 2018

Location and Time: Tuesdays 17:00-19:00, **AOB Library**

Office Hours: Any time when my office door is open;
After class is the best - ping me via e-mail.

Grading: 8-10 homeworks, 80% ; final exam: 20%
key: >85%=10, >75%=9, >65%=8, >55%=7, >45%=6 .

Class web site: <https://github.com/astromundus/astroinfo2018>

Reference textbook:

Ivezić, Connolly, VanderPlas & Gray: *Statistics, Data Mining, and Machine Learning in Astronomy: A Practical Python Guide for the Analysis of Survey Data* (Princeton University Press, 2014)
See <http://press.princeton.edu/titles/10159.html>

More (free!) resources:

- Concise “handbook”: *Notes on statistics for physicists* by Orear,
<http://www.astro.washington.edu/users/ivezic/Teaching/Astr507/orrear.pdf>
- A great book: *Probability Theory: The Logic of Science* by Jaynes,
<http://bayes.wustl.edu/etj/prob/book.pdf>
- A book about python and data science by Jake VanderPlas:
<https://github.com/jakevdp/PythonDataScienceHandbook>
- An intro class by Gordon Richards (Drexel University):
https://github.com/gtrichards/PHYS_T480
- An advanced class by Phil Marshall (Stanford University):
<https://github.com/KIPAC/StatisticalMethods>
- LSST Data Science Fellowship Program:
<https://github.com/LSSTC-DSFP/LSSTC-DSFP-Sessions>
- TED talk “The best stats you’ve ever seen” by Hans Rosling:
<http://ls.st/0dt>

Class Schedule:

- WEEK 1 : Introduction to class: syllabus, literature, astroML, python, matplotlib, GitHub.
- WEEK 2 : Introduction to statistics (probability, distributions, robust statistics, Central Limit Theorem, hypothesis testing).
- WEEK 3 : Maximum likelihood and applications in astronomy.
- WEEK 4 : Bayesian statistics and introduction to MCMC.
- WEEK 5 : Model parameter estimation and model selection.
- WEEK 6 : Time series analysis.
- WEEK 7 : Dimensionality reduction and regression.
- WEEK 8 : Density estimation and clustering.
- WEEK 9 : Unsupervised and supervised classification.
- WEEK 10 : Review
- FINAL EXAM

Homework

There will be eight to ten homeworks, assigned at the end of session on Tuesday, and due next Tuesday. They will be centered on practical work using python and designed to test the weekly progress.

We will use modern software engineering tools, GitHub and Jupyter notebooks for HW submission.

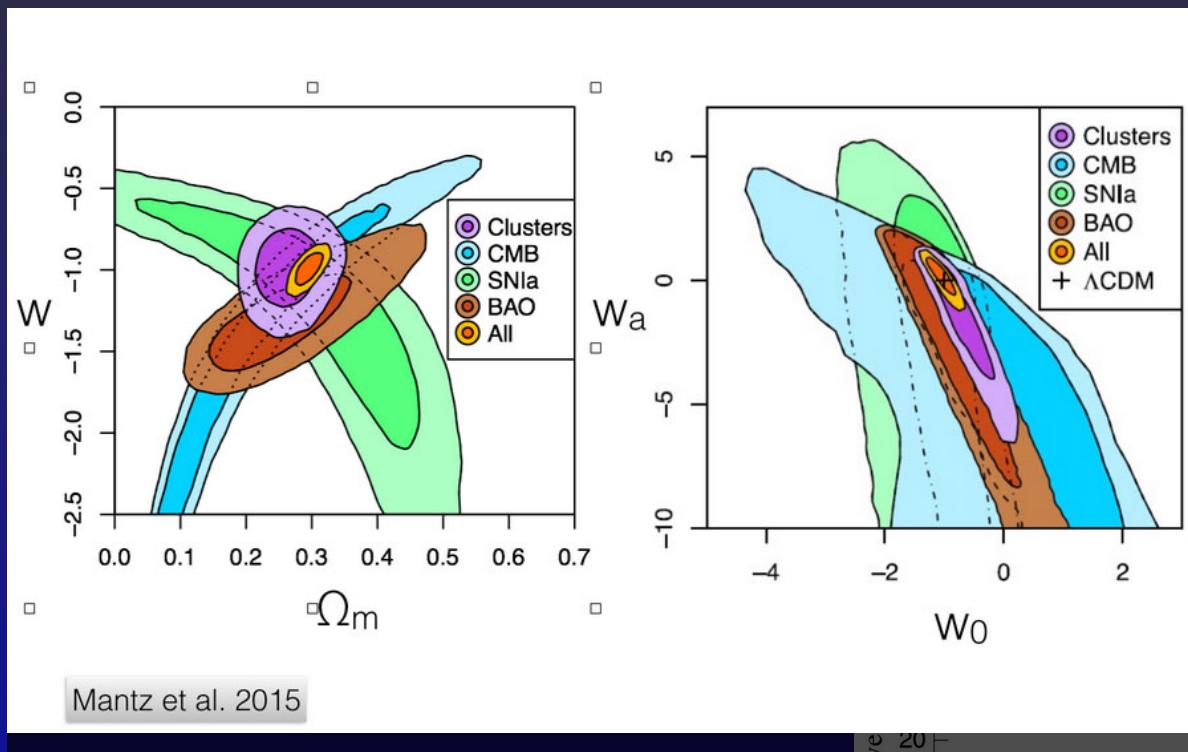
Python & Computers

- We will be using Python 3 in this class. Python has become a de-facto standard for data analysis in astronomy (and beyond)
- It's best if you had Anaconda Python installed on your laptops when following along the examples in class.
 - Will be faster than the desktops
 - You can take them home to work on homeworks.
- Macs and Linux will work. Bring your laptops along next time, we'll make sure you have everything installed.

Why Astrostatistics?

- More data, but mostly...
- More need, and the ability, to properly analyze data

Rules of thumb and approximate statistics are often insufficient...



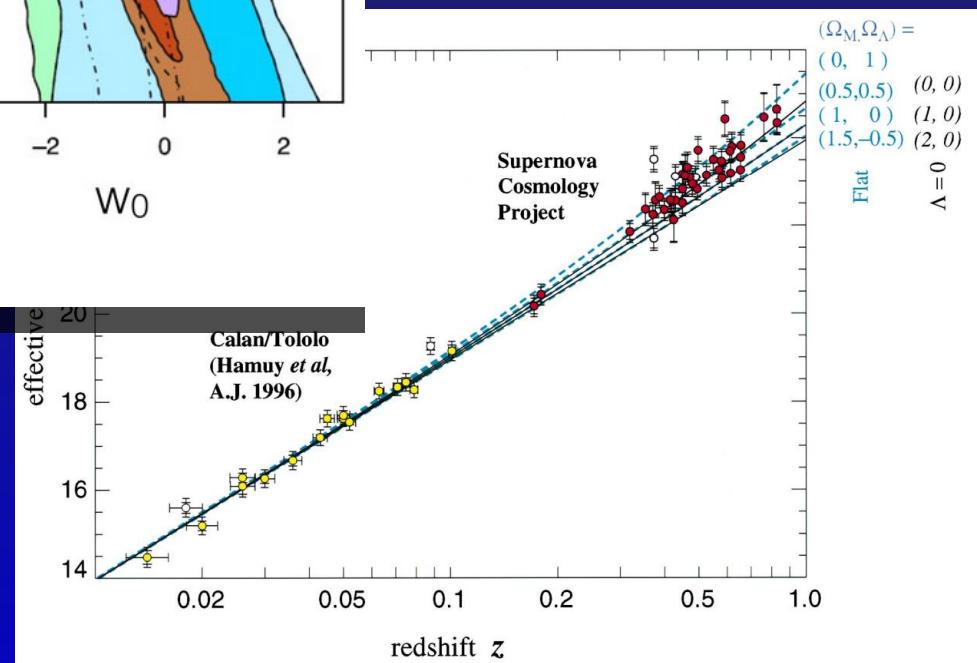
Understanding the nature of Dark Energy ↗

Discovery of Dark Energy →

Also, they used to be too (computationally) expensive, but not any more!

The signals being measured today are typically weak; careful statistical analyses are required to detect and measure them.

Because we became more sophisticated in our data analysis, today we routinely perform measurements which would've been unthinkable 20 years ago.



Graph taken from [Perlmutter et al., ApJ 517, 565 \(1999\)](#)



The first Cray supercomputer at NCSA ↗
(National Center for Supercomputing Applications)



My cell phone ↗

Big Data is growing fast



Structured and unstructured data

The digital universe will grow to
2.7ZB in 2012, up 48%

from 2011, toward nearly
8ZB by 2015¹

Americans
USE

18,264,840

MEGABYTES
OF WIRELESS DATA

YOUTUBE

USERS SHARE

400 HOURS
OF NEW VIDEO

FACEBOOK MESSENGER
USERS SHARE

216,302
PHOTOS

Amazon
MAKES

\$222,283

IN SALES

3,567,850

TEXT MESSAGES
ARE SENT

U.S.

DOMO

BUZZFEED

USERS VIEW
159,380
PIECES OF
CONTENT

SNAPCHAT

USERS WATCH
6,944,444
VIDEOS

Netflix

SUBSCRIBERS STREAM
86,805 HOURS
OF VIDEO

GOOGLE

TRANSLATES
69,500,000
WORDS

Instagram

USERS LIKE
2,430,555
POSTS

SIRI

ANSWERS
99,206
REQUESTS

Tinder

USERS SWIPE
972,222
TIMES

THE WEATHER CHANNEL

RECEIVES

13,888,889

FORECAST
REQUESTS

DOMO

YOUTUBE

USERS SHARE

400 HOURS
OF NEW VIDEO

FACEBOOK MESSENGER
USERS SHARE

216,302
PHOTOS

Amazon
MAKES

\$222,283

IN SALES

3,567,850

TEXT MESSAGES
ARE SENT

U.S.

DOMO

BUZZFEED

USERS VIEW
159,380
PIECES OF
CONTENT

SNAPCHAT

USERS WATCH
6,944,444
VIDEOS

Netflix

SUBSCRIBERS STREAM
86,805 HOURS
OF VIDEO

GOOGLE

TRANSLATES
69,500,000
WORDS

Instagram

USERS LIKE
2,430,555
POSTS

SIRI

ANSWERS
99,206
REQUESTS

Tinder

USERS SWIPE
972,222
TIMES

THE WEATHER CHANNEL

RECEIVES

13,888,889

FORECAST
REQUESTS

DOMO

2016
every
MINUTE
of
the
DAY

PRESNTED BY DOMO

Giphy
SERVES
569,217
GIFS

TWITTER
USERS SEND
9,678
EMOJI-FILLED TWEETS

Dropbox
USERS UPLOAD
833,333
NEW FILES

DATA NEVER SLEEPS 4.0

How much data is generated every minute? In the fourth annual edition of Data Never Sleeps, newcomers like Giphy and Facebook Messenger illustrate the rise of our multimedia messaging obsession, while veterans like YouTube and Snapchat highlight our insatiable appetite for video. Just how many GIFs, videos, and emoji-filled Tweets flood the internet every minute? See for yourself below.

| Sky Survey Projects | Data Volume |
|--|-------------------|
| DPOSS (The Palomar Digital Sky Survey) | 3 TB |
| 2MASS (The Two Micron All-Sky Survey) | 10 TB |
| GBT (Green Bank Telescope) | 20 PB |
| GALEX (The Galaxy Evolution Explorer) | 30 TB |
| SDSS (The Sloan Digital Sky Survey) | 40 TB |
| SkyMapper Southern Sky Survey | 500 TB |
| PanSTARRS (The Panoramic Survey Telescope and Rapid Response System) | ~ 40 PB expected |
| LSST (The Large Synoptic Survey Telescope) | ~ 200 PB expected |
| SKA (The Square Kilometer Array) | ~ 4.6 EB expected |



LSST in one sentence:

An optical/near-IR survey of half the sky in ugrizy bands to $r \sim 27.5$ based on ~ 800 visits over a 10-year period:

A catalog of 20 billion stars and 20 billion galaxies with exquisite photometry, astrometry and image quality!

3.6×10^{-31}
erg/s/cm²/Hz

More information at
www.lsst.org
and arXiv:0805.2366

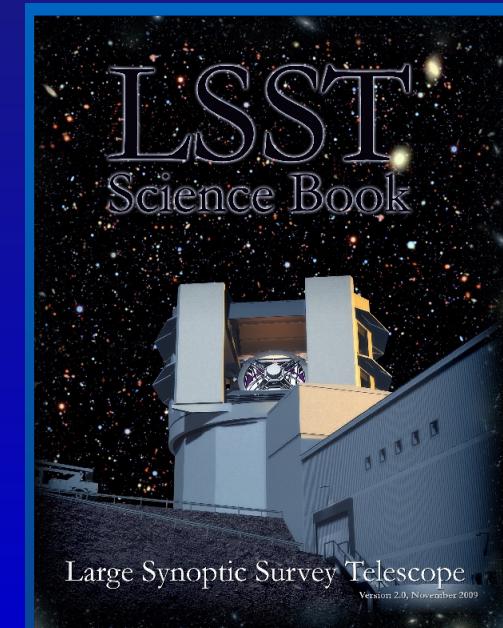
LSST Science Themes

- Dark matter, dark energy, cosmology
(spatial distribution of galaxies,
gravitational lensing, supernovae, quasars)
- Time domain (cosmic
explosions, variable stars)
- The Solar System structure (asteroids)
- The Milky Way structure (stars)

LSST Science Book: arXiv:0912.0201

Summarizes LSST hardware, software, and observing plans, science enabled by LSST, and educational and outreach opportunities

245 authors, 15 chapters, 600 pages

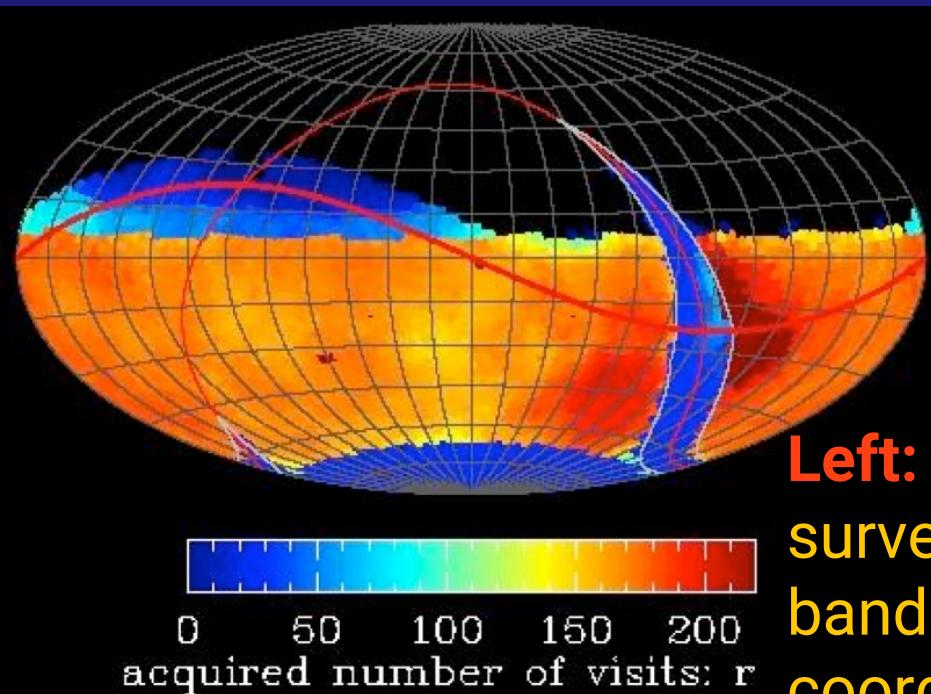


Large Synoptic Survey Telescope

Version 3.0, November 2009

Basic idea behind LSST: a uniform sky survey

- 90% of time will be spent on a uniform survey: every 3-4 nights, the whole observable sky will be scanned twice per night
- after 10 years, half of the sky will be imaged about 1000 times (in 6 bandpasses, ugrizy): a digital color movie of the sky
- ~100 PB of data: about a billion 16 Mpix images, enabling **measurements for 40 billion objects!**



LSST in one sentence:

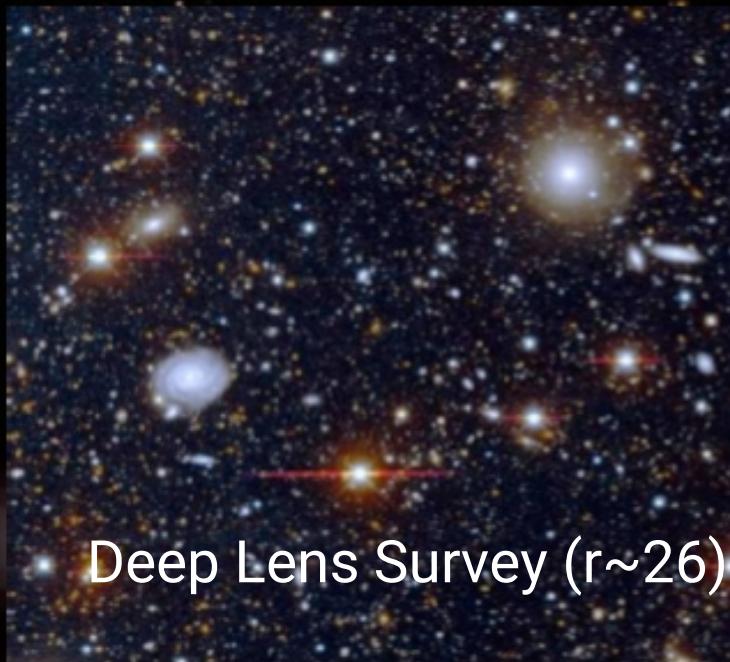
An optical/near-IR survey of half the sky in ugrizy bands to $r \sim 27.5$ (36 nJy) based on 825 visits over a 10-year period: deep wide fast.

Left: a 10-year simulation of LSST survey: the number of visits in the r band (Aitoff projection of eq. coordinates)

SDSS vs. LSST comparison: LSST=d(SDSS)/dt,

LSST=SuperSDSS
3x3 arcmin, gri

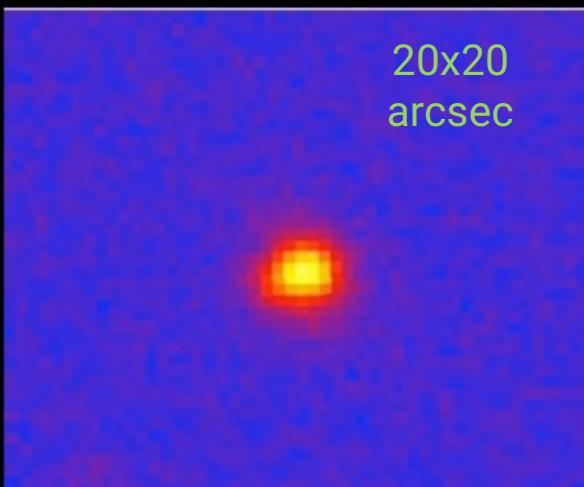
3 arcmin
is 1/10 of
the full
Moon's
diameter



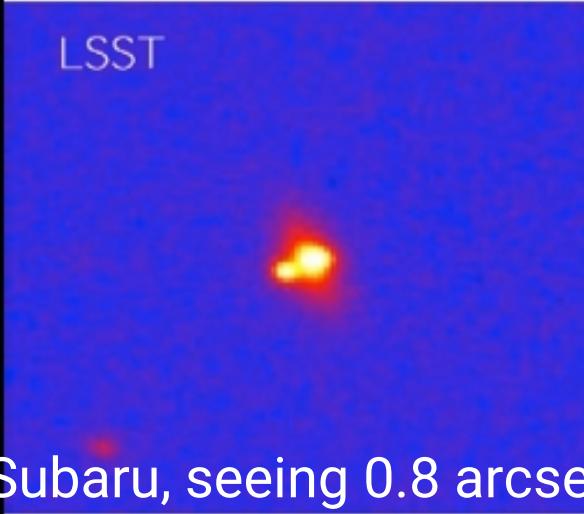
Deep Lens Survey ($r \sim 26$)

(almost)
like LSST
depth
(but tiny
area)

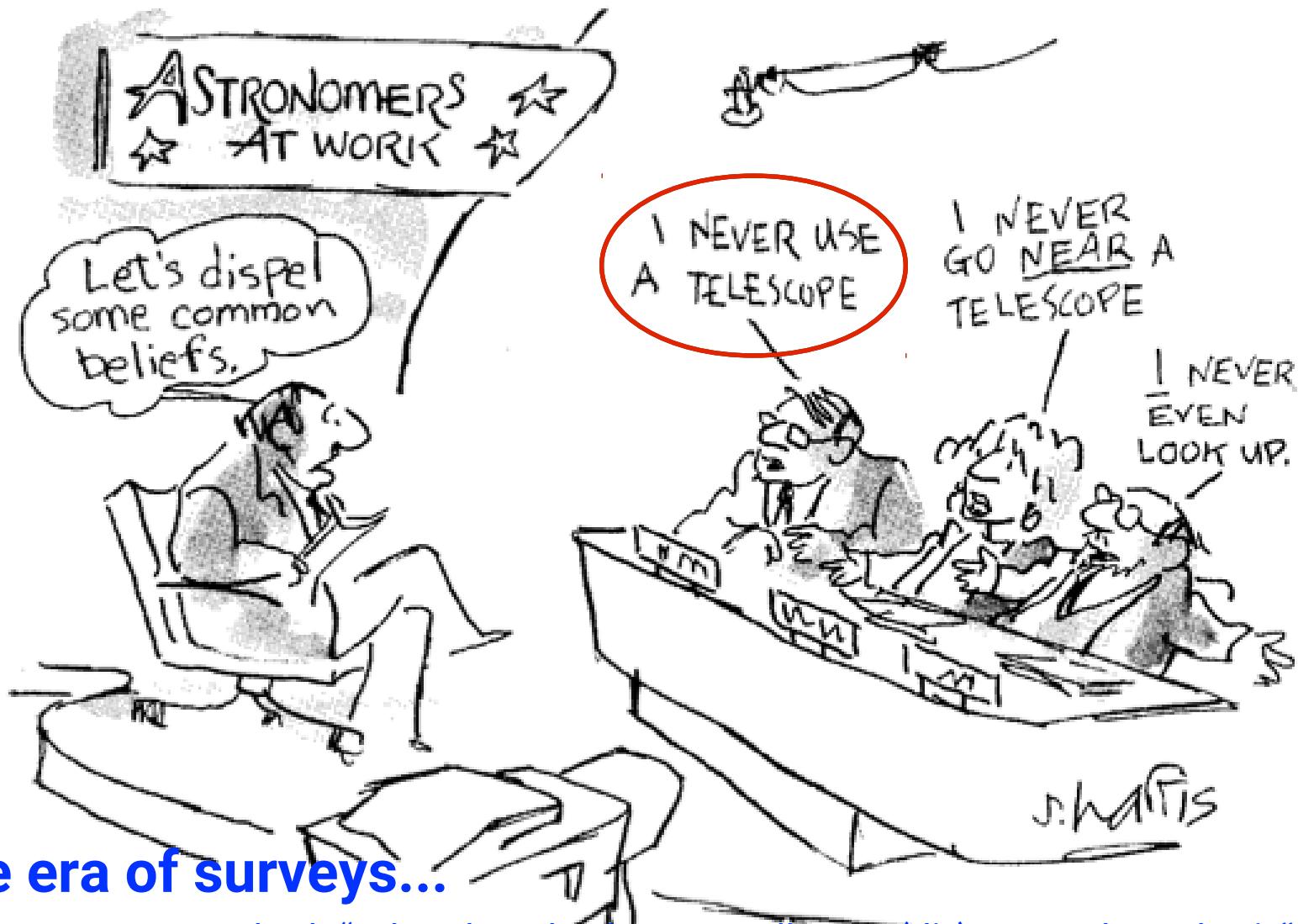
20x20 arcsec; lensed SDSS quasar
(SDSS J1332+0347, Morokuma et al. 2007)



SDSS, seeing 1.5 arcsec



Subaru, seeing 0.8 arcsec



The era of surveys...

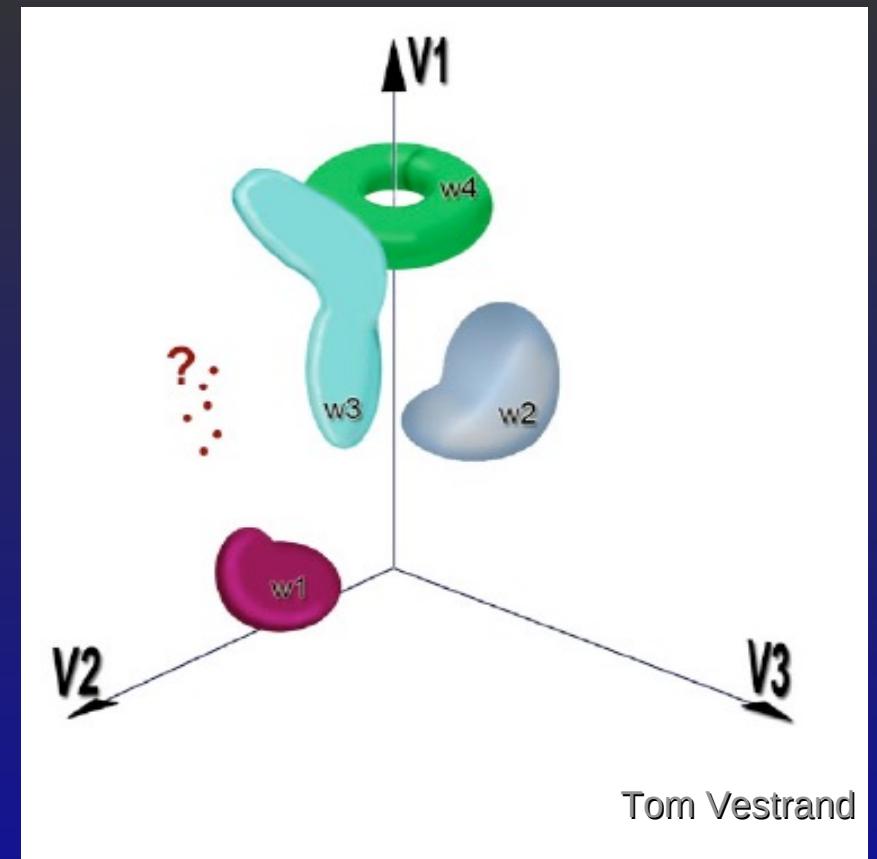
- Standard: “What data do I have to collect to (dis)prove a hypothesis”?
- Data-driven: “What theories can I test given the data I already have?”

Data analysis challenges in the era of Big Data

The bottleneck is not any more data availability but instead our ability to extract useful and reliable information from data.

- 1) Large data volume (petabytes)
- 2) Large numbers of objects (billions)
- 3) Highly multi-dimensional spaces (thousands)
- 4) Unknown statistical distributions
- 5) Time-series data (irregular sampling)
- 6) Heteroscedastic errors, truncated, censored and missing data
- 7) Unreliable quantities (e.g. unknown systematics and random errors)

- ▲ Characterize the known clustering)
- ▲ Assign the new (classification)
- ▲ Discover the unknown (outlier detection)



Benefits of very large data sets:

- best statistical analysis of “typical” events
- automated search for “rare” events

**In this class, you will learn how to do all that.
Btw, it need not be an astronomical application!**



Alternative Careers: Leveraging your Astronomy Degree for Data Science

by Ben Cook | Jun 1, 2016 | Career Navigation, Personal Experiences | 0 comments

Big Data in Astronomy

Alongside the recent explosion of “[Big Data](#)” into the public consciousness, there has been a similar transition into the age of “[Big Astronomy](#)”. Astronomers have always been adept at drawing conclusions using [advanced statistics](#) and [data analysis](#). Now, with the advent of extremely large simulations like [Illustris](#) and surveys like the upcoming LSST, astronomers are increasingly gaining experience in dealing with [datasets vastly larger](#) than could ever hope to fit on a single computer.

For early career astronomers looking for advice, I think you can do no better than look at the posts made by Jessica Kirkpatrick, who obtained a PhD in Astronomy and then became a data scientist at Microsoft/Yammer, and I understand she has since taken a position as Director of Data Science at the education start-up [InstaEDU](#).

The term “Data Scientist” is extraordinarily broad. For example, the post “[What is a Data Scientist?](#)” describes some of the Data Analyst roles a Data Scientists may play:

- Derive business insight from data.
- Work across all teams within an organization.
- Answer questions using analysis of data.
- Design and perform experiments and tests.
- Create forecasts and models.
- Prioritize which questions and analyses are actionable and valuable.
- Help teams/executives make data-driven decisions.
- Communicate results across the company to technical and non-technical people.

What to expect from this class

- **This class is not a computer science class**
- **This class is not a statistics class**
- **This class is not an astronomy class**
- The main purpose of this class is to expose you to a combination of concepts and tools from the above fields so that you can “handle” your “data” (as in data & noise & models & understanding etc.). We want to demystify statistical methods and give you the practical tools to apply them.
- “Your data” can be big and small, astronomical or not, but most basic concepts and tools will have a general utility (e.g. histograms - which will be much more fun than you think!)

AstroML (and Visualization)

News

October 2012: astroML 0.1 has been released! Get the source on [Github](#)

Our Introduction to astroML paper received the CIDU 2012 best paper award.

Links

[astroML Mailing List](#)

[GitHub Issue Tracker](#)

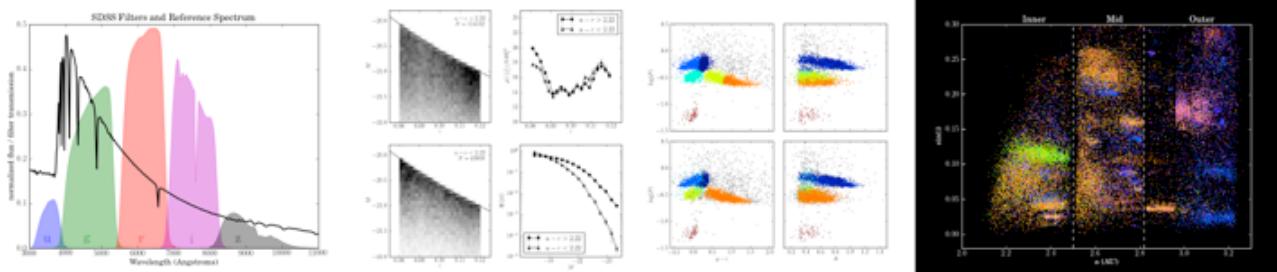
Videos

[Scipy 2012 \(15 minute talk\)](#)

Citing

If you use the software, please consider citing astroML.

AstroML: Machine Learning and Data Mining for Astronomy

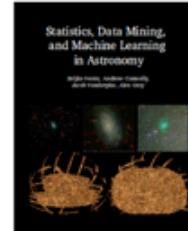


AstroML is a Python module for machine learning and data mining built on [numpy](#), [scipy](#), [scikit-learn](#), and [matplotlib](#), and distributed under the 3-clause BSD license. It contains a growing library of statistical and machine learning routines for analyzing astronomical data in python, loaders for several open astronomical datasets, and a large suite of examples of analyzing and visualizing astronomical datasets.

The goal of astroML is to provide a community repository for fast Python implementations of common tools and routines used for statistical data analysis in astronomy and astrophysics, to provide a uniform and easy-to-use interface to freely available astronomical datasets. We hope this package will be useful to researchers and students of astronomy. The astroML project was started in 2012 to accompany the book **Statistics, Data Mining, and Machine Learning in Astronomy** by Zeljko Ivezic, Andrew Connolly, Jacob VanderPlas, and Alex Gray, to be published in late 2013. The table of contents is available here: [here \(pdf\)](#).

Downloads

- Released Versions: [Python Package Index](#)
- Bleeding-edge Source: [github](#)



User Guide

1. Introduction

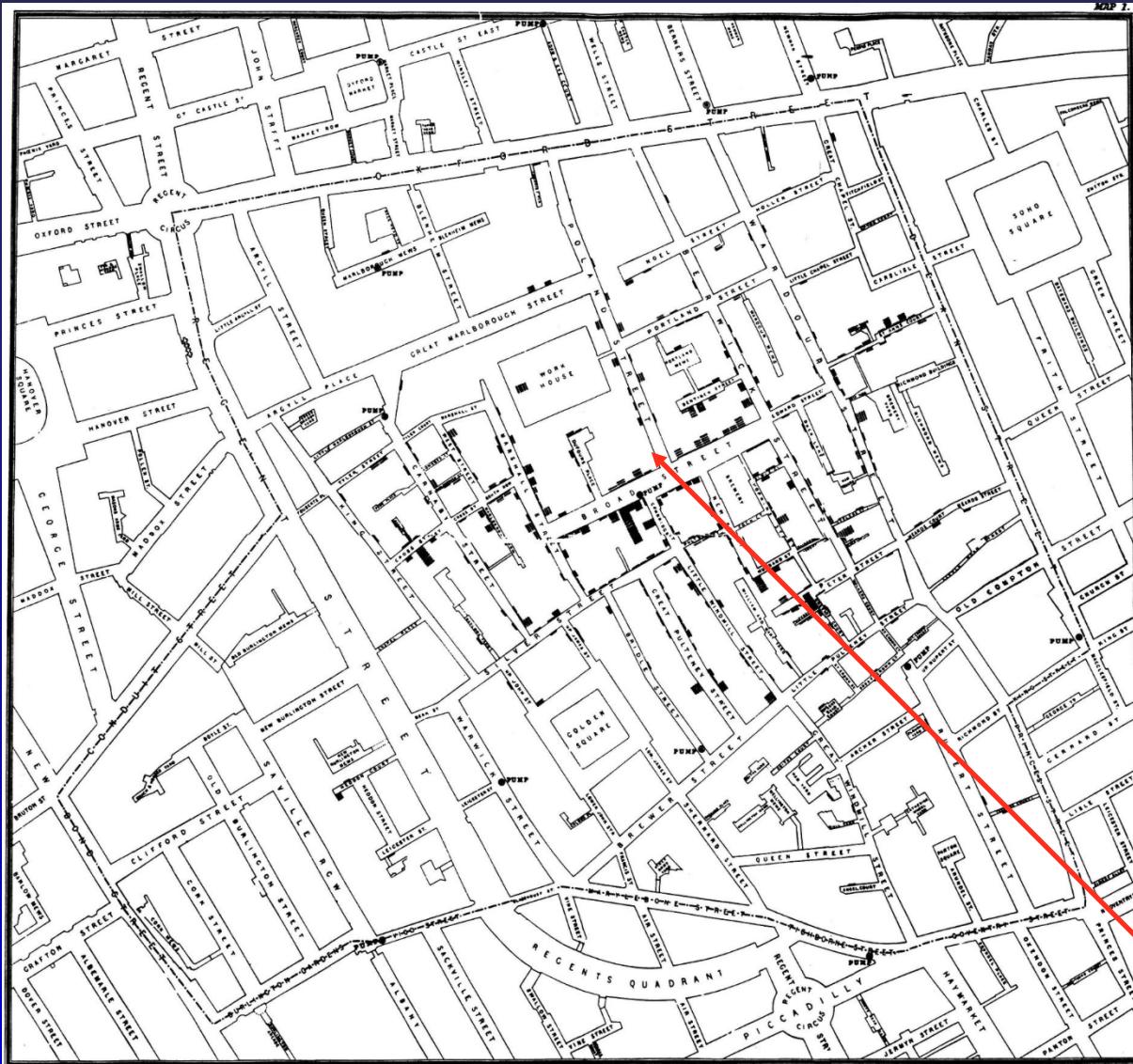
- 1.1. Philosophy

Open source!

www.astroml.org

Visualization of multi-dimensional correlations and search for patterns

“The greatest value of a picture is when it forces us to notice what we never expected to see.” (John Tukey, 1977)



John Snow plotted the number of cholera deaths (black marks) in London in 1854 outbreak and realized that the deaths are clustered around the Broad Street pump: the cholera is spread by water!

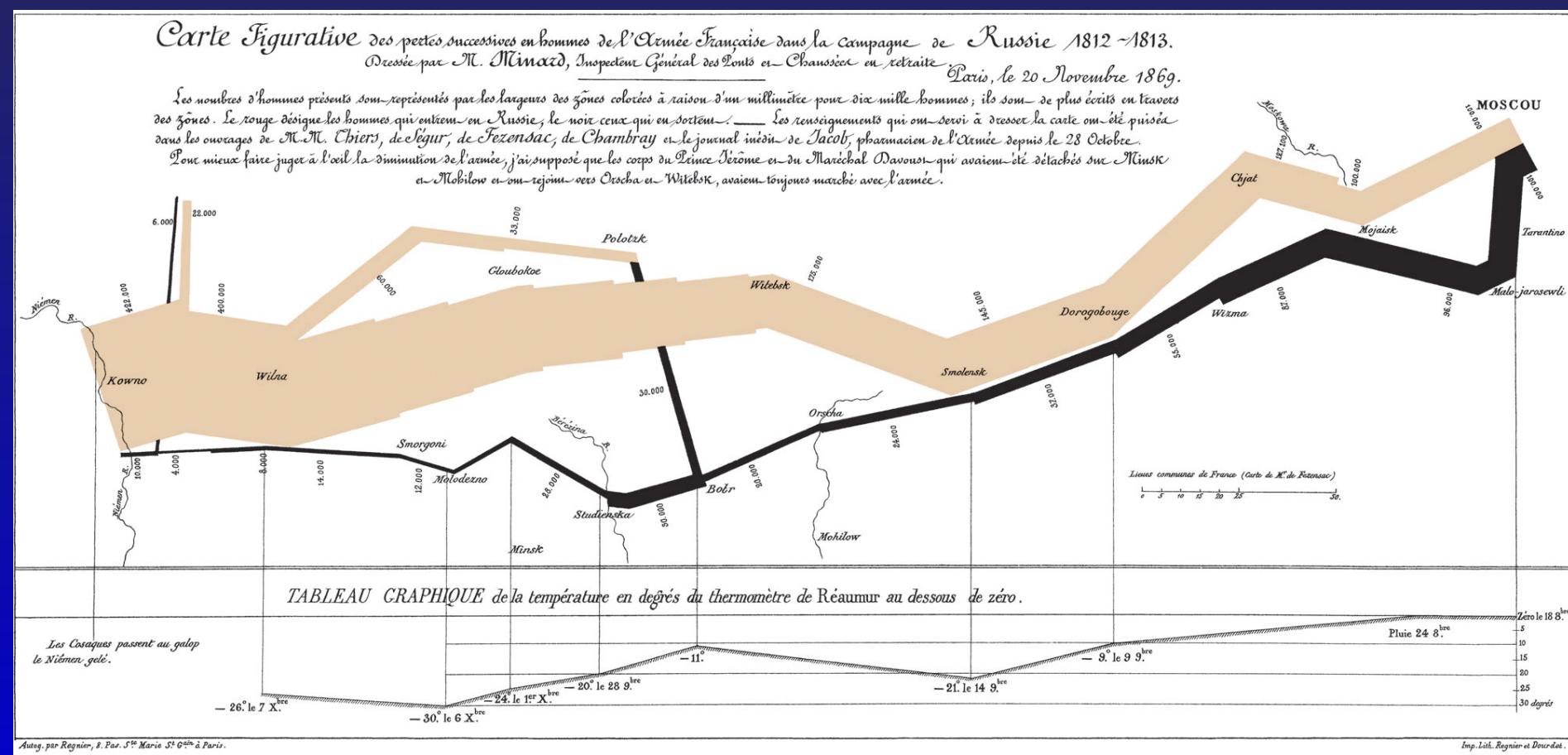
https://en.wikipedia.org/wiki/1854_Broad_Street_cholera_outbreak

The Broad Street pump

Visualization of multi-dimensional correlations and search for patterns

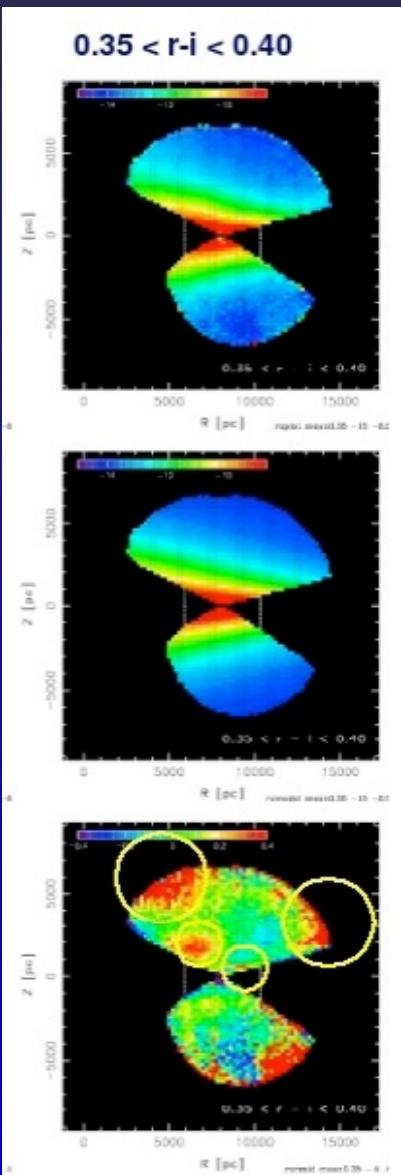
Napoleon's march on Moscow: 4-D data (2 positions, time, army size; also temperature and some other "metadata")

98% of soldiers perished... (visualization by Charles Minard)



Visualization of multi-dimensional correlations and search for patterns

“The greatest value of a picture is when it forces us to notice what we never expected to see.” (John Tukey, 1977)

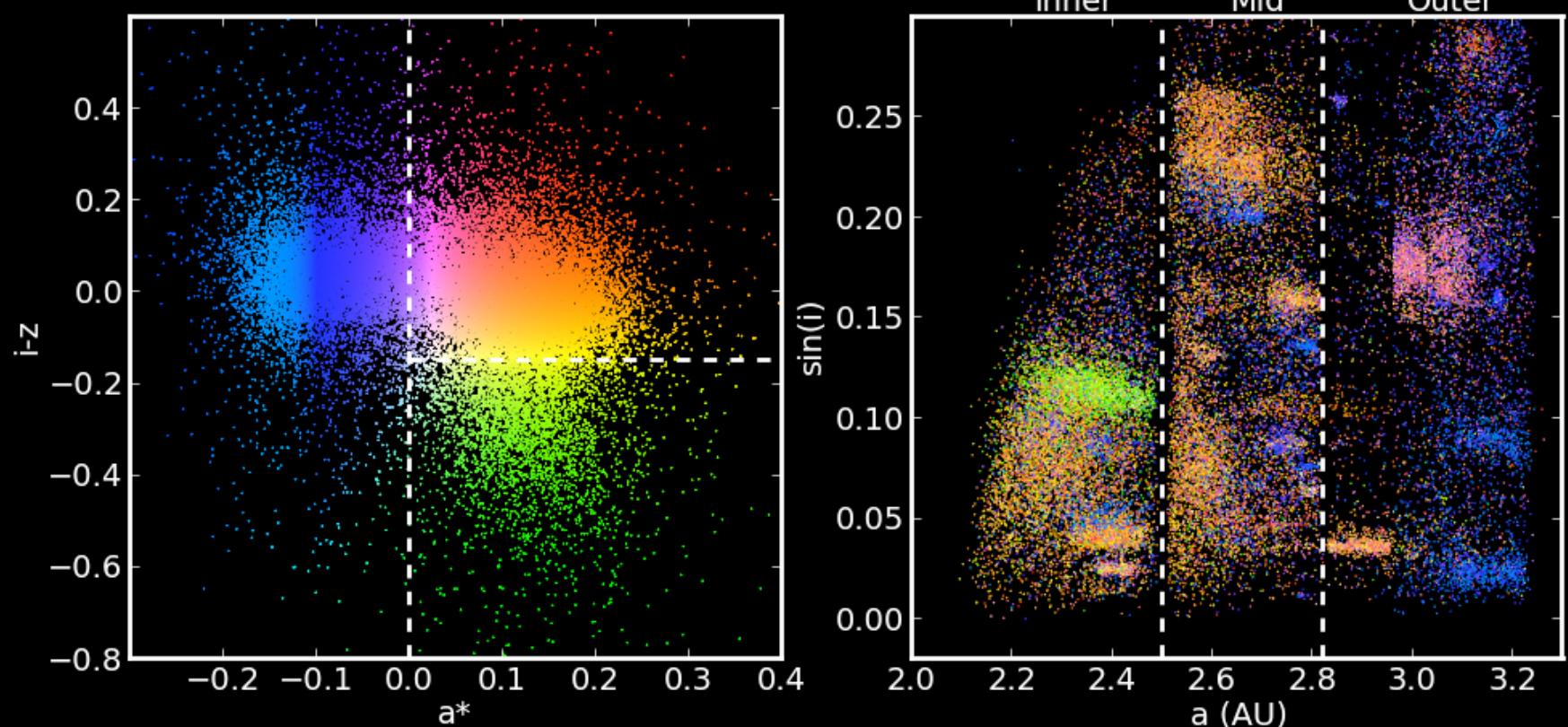


- An example from Jurić et al. (2008): our Galaxy grew by cannibalizing nearby smaller neighbors
- Four-dimensional data: stellar color, apparent magnitude, position on the sky (RA, Dec)
- Step 1: derive three dimensional positions and bin the data in cylindrical coordinates
- Step 2: fit a (relatively complicated) model
- Step 3: subtract the best-fit model from data
- We will learn about all these steps in this class.

You will be able to make this plot yourself!

Ivezic et al. (2002)

Visualization of 4-dimensional correlations



Homework #1 is based on this example

[Previous
Stellar
Paramete...
...](#) [Next
Mercator
Project...
...](#) [Up
Chapter 1:
Intro...](#)

This documentation is for
astroML version 0.2

This page

SDSS Stripe 82 Moving Object Catalog

Links

[astroML Mailing List](#)

[GitHub Issue Tracker](#)

Videos

[Scipy 2012 \(15 minute talk\)](#)

[Scipy 2013 \(20 minute talk\)](#)

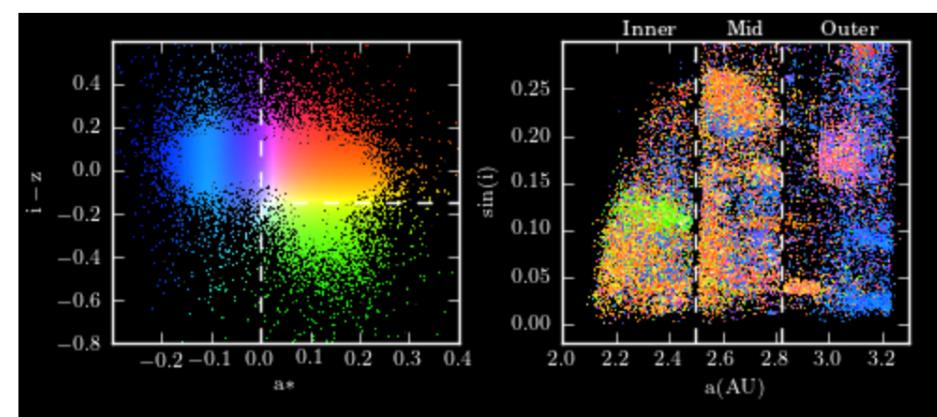
Citing

If you use the software,
please consider citing
astroML.

SDSS Stripe 82 Moving Object Catalog

Figure 1.12.

A multicolor scatter plot of the properties of asteroids from the SDSS Moving Object Catalog (cf. figure 1.8). The left panel shows observational markers of the chemical properties of the asteroids: two colors a^* and $i-z$. The right panel shows the orbital parameters: semimajor axis a vs. the sine of the inclination. The color of points in the right panel reflects their position in the left panel. This plot is similar to that used in figures 3-4 of Parker et al 2008.



▶ Code output:

▼ Python source code:

```
# Author: Jake VanderPlas
# License: BSD
# The figure produced by this code is published in the textbook
# "Statistics, Data Mining, and Machine Learning in Astronomy" (2013)
# For more information, see http://astroml.github.com
# To report a bug or issue, use the following forum:
# https://groups.google.com/forum/#!forum/astroml-general
import numpy as np
from matplotlib import pyplot as plt
```

Textbook Figures

This section makes available the source code used to generate every figure in the book *Statistics, Data Mining, and Machine Learning in Astronomy*. Many of the figures are fairly self-explanatory, though some will be less so without the book as a reference. The table of contents of the book can be seen [here \(pdf\)](#).

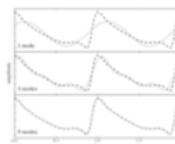
Figure Contents

Each chapter links to a page with thumbnails of the figures from the chapter.

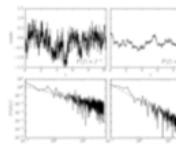
- Chapter 1: Introduction
- Chapter 2: Fast Computation and Massive Datasets
- Chapter 3: Probability and Statistical Distributions
- Chapter 4: Classical Statistical Inference
- Chapter 5: Bayesian Statistical Inference
- Chapter 6: Searching for Structure in Point Data
- Chapter 7: Dimensionality and its Reduction
- Chapter 8: Regression and Model Fitting
- Chapter 9: Classification
- Chapter 10: Time Series Analysis
- Appendix

Chapter 10: Time Series Analysis

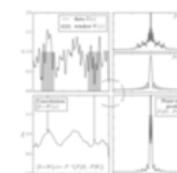
This chapter covers the analysis of both periodic and non-periodic time series, for both regularly and irregularly spaced data.



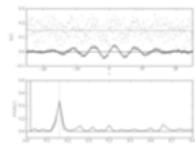
Fourier Reconstruction of
RR-Lyrae Templates



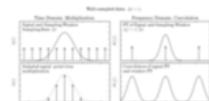
Generating Power-law
Light Curves



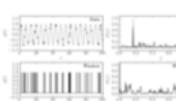
Plot a Diagram explaining
a Convolution



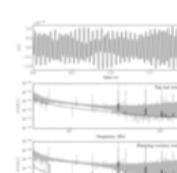
Fast Fourier Transform
Example



The effect of Sampling



The effect of Sampling



Examples of Wavelets

Plot the power spectrum of
the LIGO big dog event

- https://github.com/astroML/astroML_figures
- \$BOOK/book_figures/chapter1/fig_moving_objects_multicolor.py

```

def compute_color(mag_a, mag_i, mag_z, a_crit=-0.1):
    """
    Compute the scatter-plot color using code adapted from
    TCL source used in Parker 2008.
    """

    # define the base color scalings
    R = np.ones_like(mag_i)
    G = 0.5 * 10 ** (-2 * (mag_i - mag_z - 0.01))
    B = 1.5 * 10 ** (-8 * (mag_a + 0.0))

    # enhance green beyond the a_crit cutoff
    G += 10. / (1 + np.exp((mag_a - a_crit) / 0.02))

    # normalize color of each point to its maximum component
    RGB = np.vstack([R, G, B])
    RGB /= RGB.max(0)

    # return an array of RGB colors, which is shape (n_points, 3)
    return RGB.T

#-----
# Fetch data and extract the desired quantities
data = fetch_moving_objects(Parker2008_cuts=True)
mag_a = data['mag_a']
mag_i = data['mag_i']
mag_z = data['mag_z']
a = data['aprime']
sini = data['sin_iprime']

# dither: magnitudes are recorded only to +/- 0.01
mag_a += -0.005 + 0.01 * np.random.random(size=mag_a.shape)
mag_i += -0.005 + 0.01 * np.random.random(size=mag_i.shape)
mag_z += -0.005 + 0.01 * np.random.random(size=mag_z.shape)

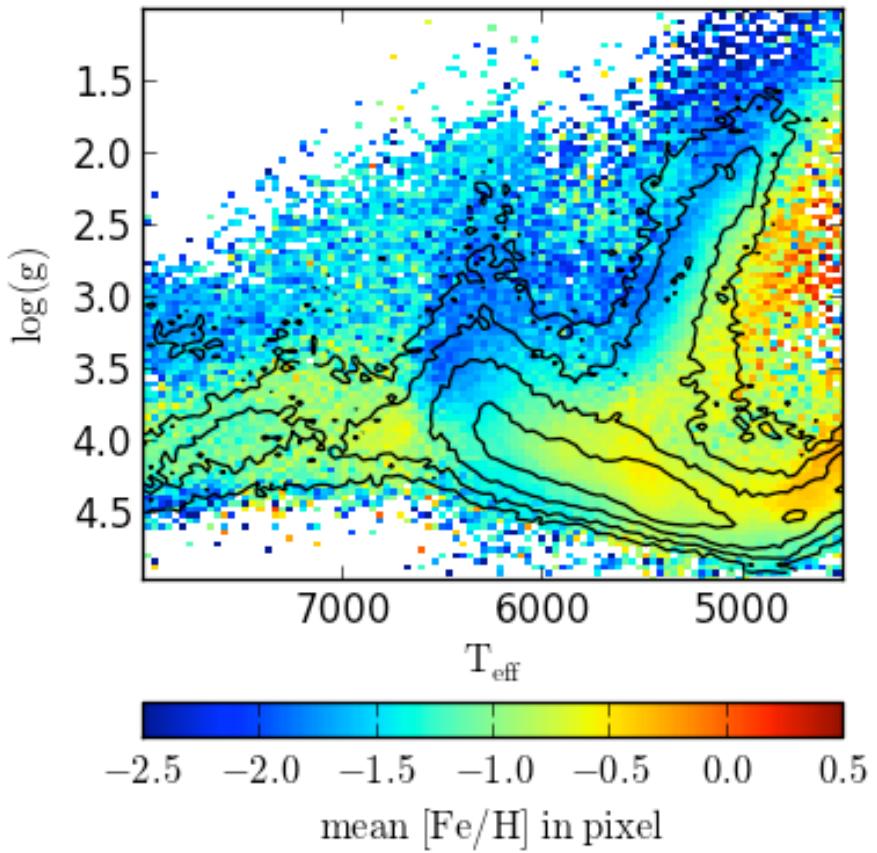
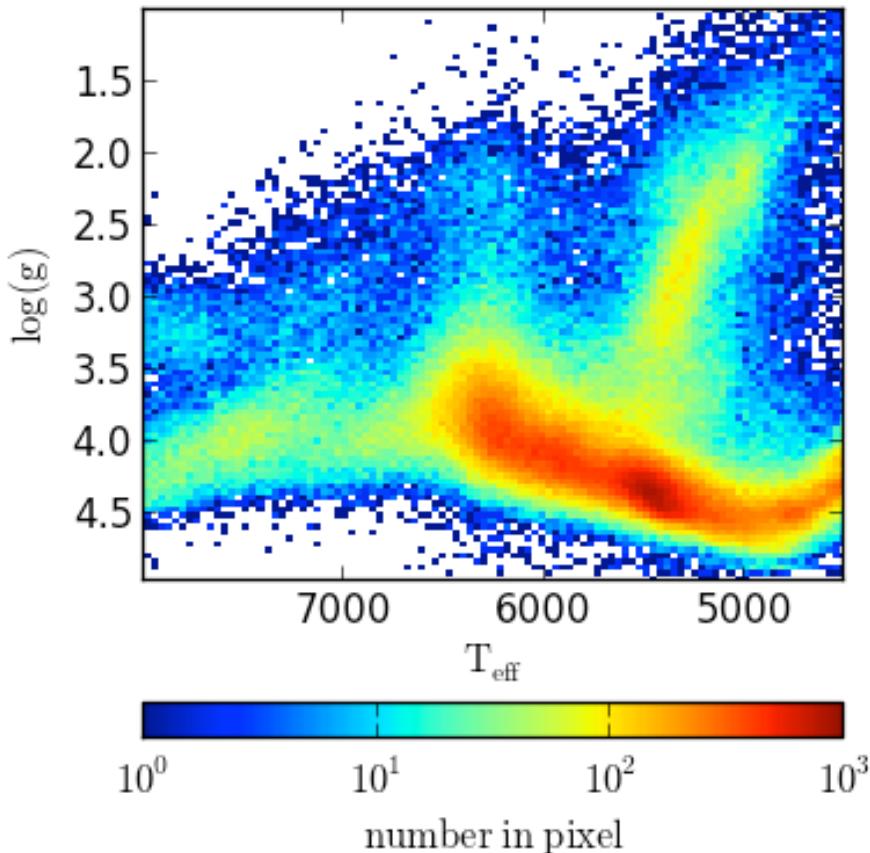
# compute RGB color based on magnitudes
color = compute_color(mag_a, mag_i, mag_z)

#-----
# set up the plot
fig = plt.figure(figsize=(5, 2.2), facecolor='k')
fig.subplots_adjust(left=0.1, right=0.95, wspace=0.3,
                    bottom=0.2, top=0.93)

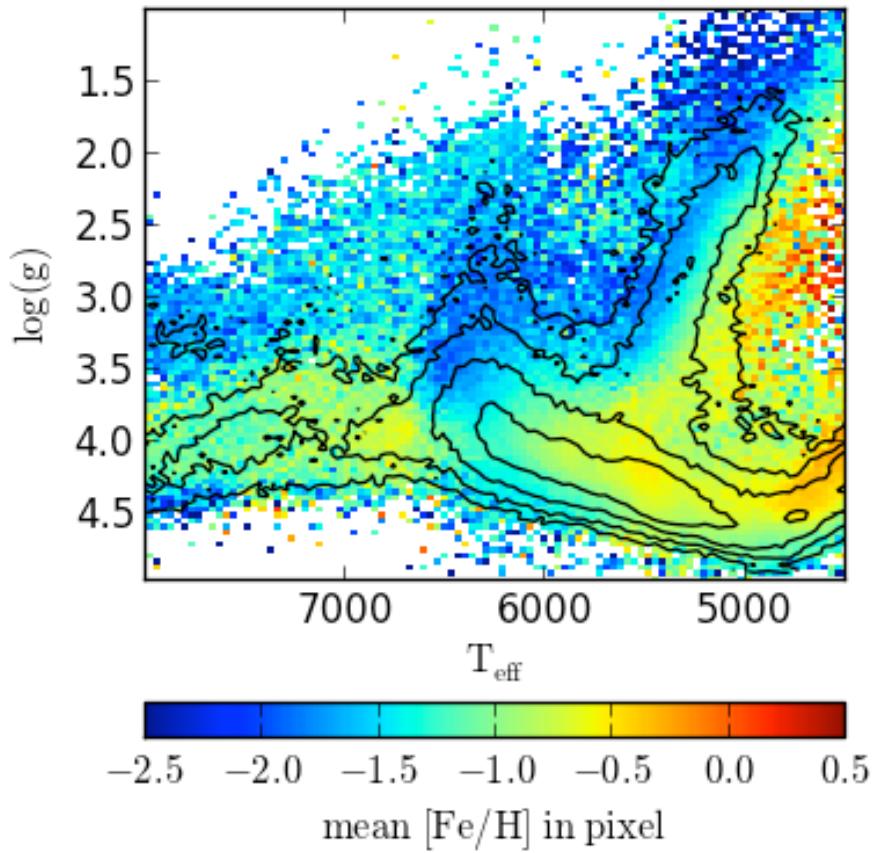
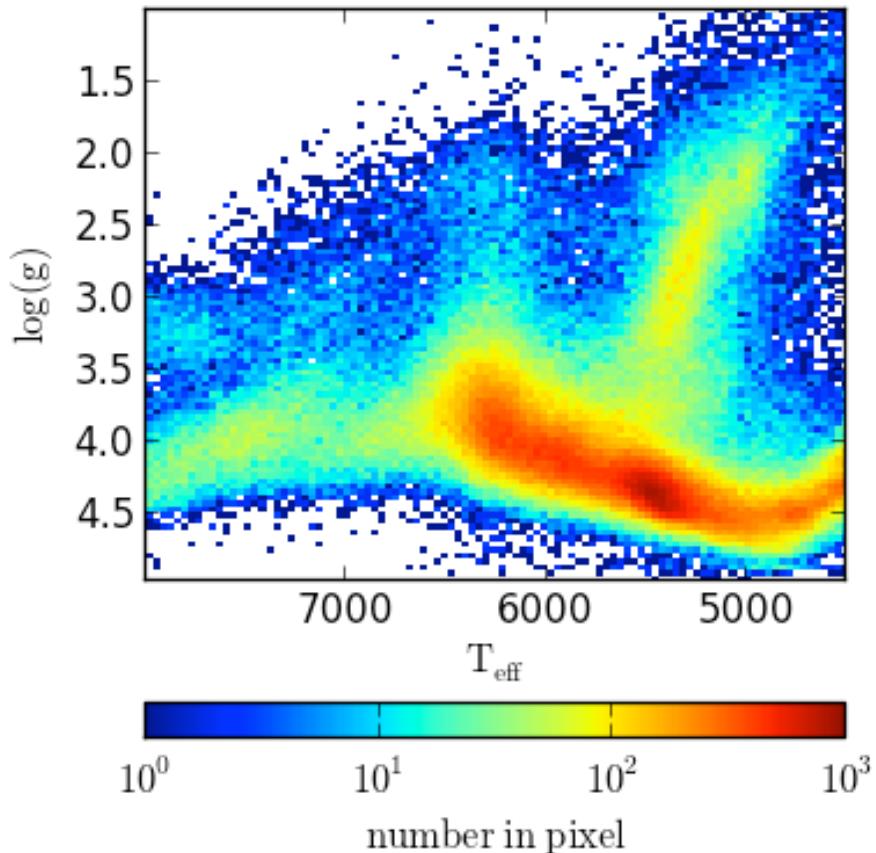
```

- A simple example of astroML use:
a Hess
diagram coded by a third quantity
- Hess diagram is astronomical term for pixelated
color-magnitude diagram, where each pixel is coded
to display the number of objects in it (also known as
“two-dimensional histogram”)

- **A Hess diagram coded by a third quantity**
- Here we plot a measure of the star's surface gravity strength vs. effective temperature (both estimated from a spectrum obtained by SDSS); the left panel shows the count of stars in each pixel



- Of course, the pixels don't have to be coded by the number of objects in it - we can use instead any other statistic (the mean, median, scatter, etc): the right panel shows the mean metallicity [Fe/H] by color and the same counts as in the left panel, but now using contours



- \$astroMLdir/astroML/stats/_binned_statistic.py

```
def binned_statistic(x, values, statistic='mean',
                     bins=10, range=None):
```

Compute a binned statistic for a set of data.

This is a generalization of a histogram function. A histogram divides the space into bins, and returns the count of the number of points in each bin. This function allows the computation of the sum, mean, median, or other statistic of the values within each bin.

Parameters

x : array_like

A sequence of values to be binned.

values : array_like

The values on which the statistic will be computed. This must be the same shape as x.

statistic : string or callable, optional

The statistic to compute (default is 'mean').

The following statistics are available:

- * 'mean' : compute the mean of values for points within each bin.
Empty bins will be represented by NaN.
- * 'median' : compute the median of values for points within each bin. Empty bins will be represented by NaN.
- * 'count' : compute the count of points within each bin. This is identical to an unweighted histogram. 'values' array is not referenced.
- * 'sum' : compute the sum of values for points within each bin.
This is identical to a weighted histogram.
- * function : a user-defined function which takes a 1D array of values, and outputs a single numerical statistic. This function will be called on the values in each bin. Empty bins will be represented by function([]), or NaN if this returns an error.

bins : int or sequence of scalars, optional

If `bins` is an int, it defines the number of equal-width

- \$astroMLdir/astroML/stats/_binned_statistic.py

```
def binned_statistic(x, values, statistic='mean',
                     bins=10, range=None):
    """Compute a binned statistic for a set of data.

    This is a generalization of a histogram function. A histogram divides
    the space into bins, and returns the count of the number of points in
    each bin. This function allows the computation of the sum, mean, median,
    or other statistic of the values within each bin.

    Parameters
    -----
    x : array_like
        A sequence of values to be binned.
    values : array_like
```

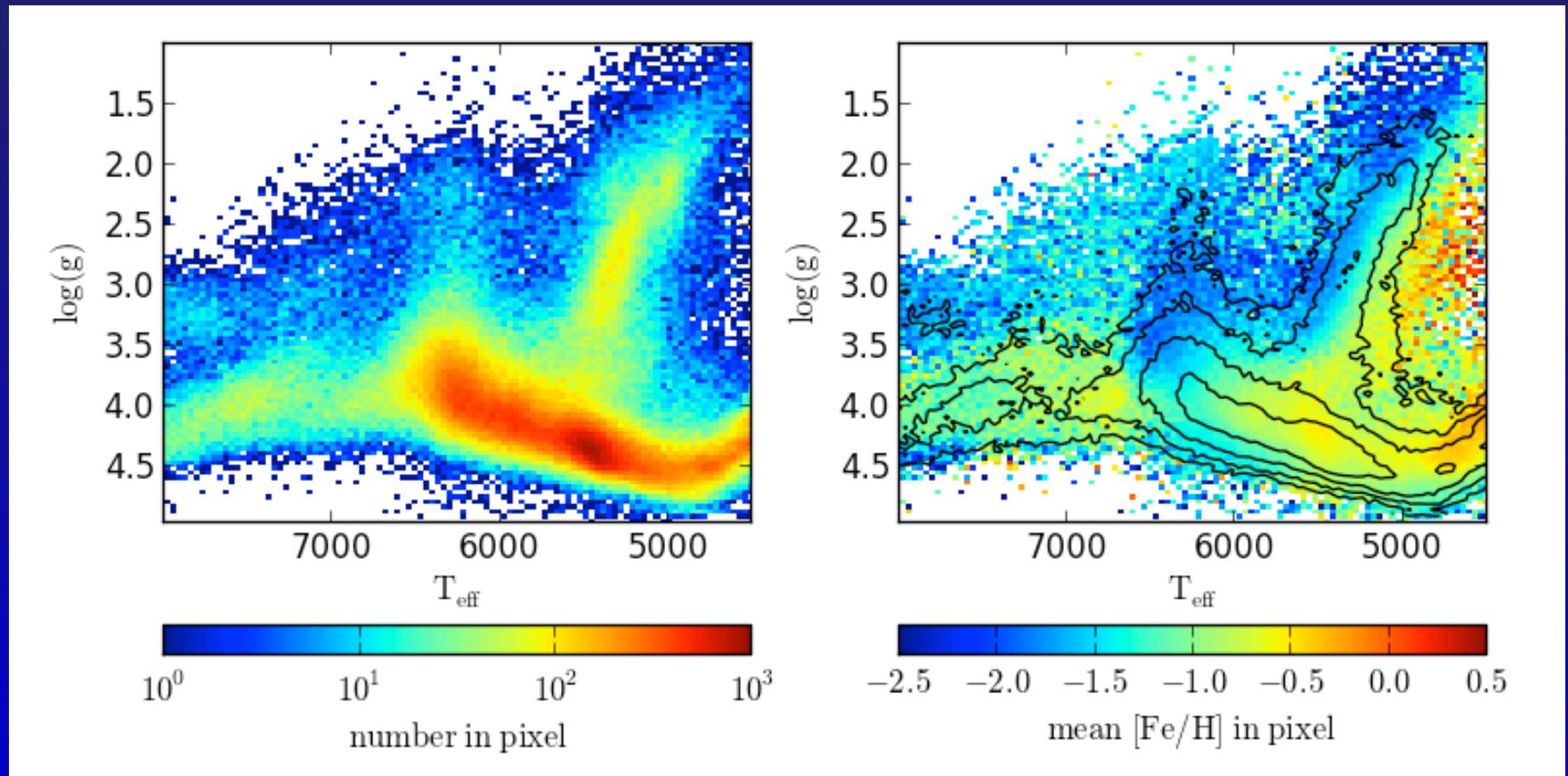
```
#-----
# Plot the results using the binned_statistic function
from astroML.stats import binned_statistic_2d
N, xedges, yedges = binned_statistic_2d(Teff, logg, FeH,
                                         'count', bins=100)

FeH_mean, xedges, yedges = binned_statistic_2d(Teff, logg, FeH,
                                                'mean', bins=100)
```

- \$astroMLdir/astroML/stats/_binned_statistic.py

```
bins : int or sequence of scalars, optional
       If `bins` is an int, it defines the number of equal-width
```

- make this plot by running
`%run $astroMLdir/examples/datasets/plot_SDSS_SSPP.py`
- now, let's look at the code and change something, for example, let's only select stars with $15 < \text{rpsf} < 16$
 find line 23: `data = data[(rpsf > 15) & (rpsf < 19)]`
 and change 19 to 16, save and then rerun...



- make this plot by running
`%run $astroMLdir/examples/datasets/plot_SDSS_SSPP.py`
- now, let's look at the code and change something, for example, let's only select stars with $15 < \text{rpsf} < 16$
 find line 23: `data = data[(rpsf > 15) & (rpsf < 19)]`
 and change 19 to 16, save and then rerun...

